

FIG. 1

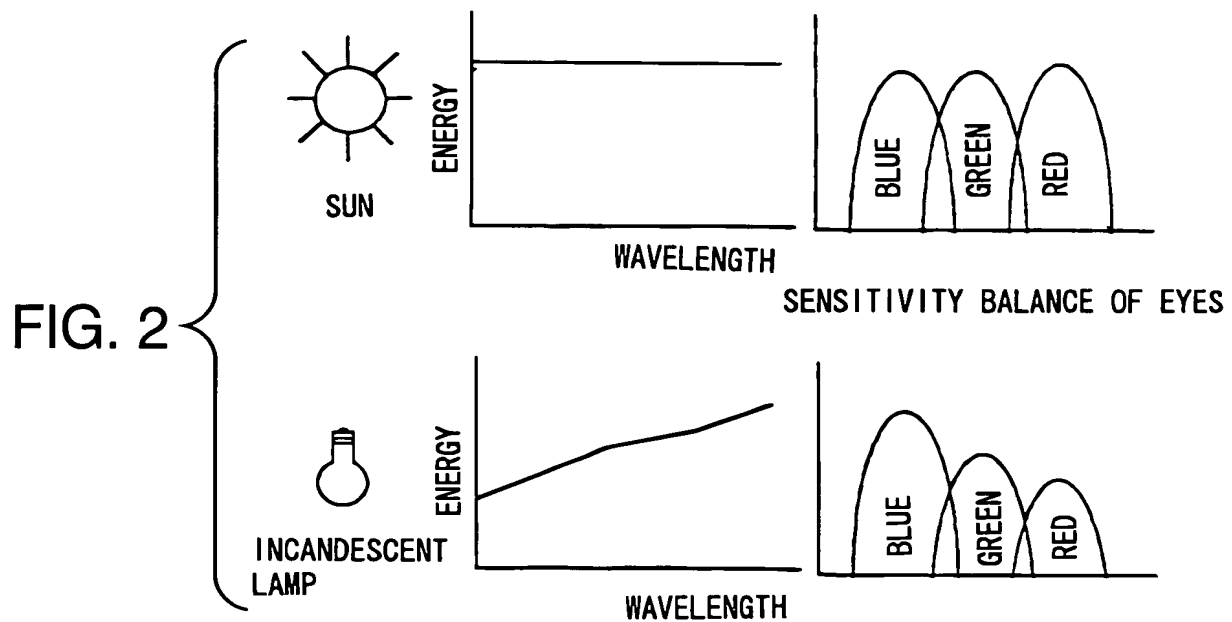


FIG. 3

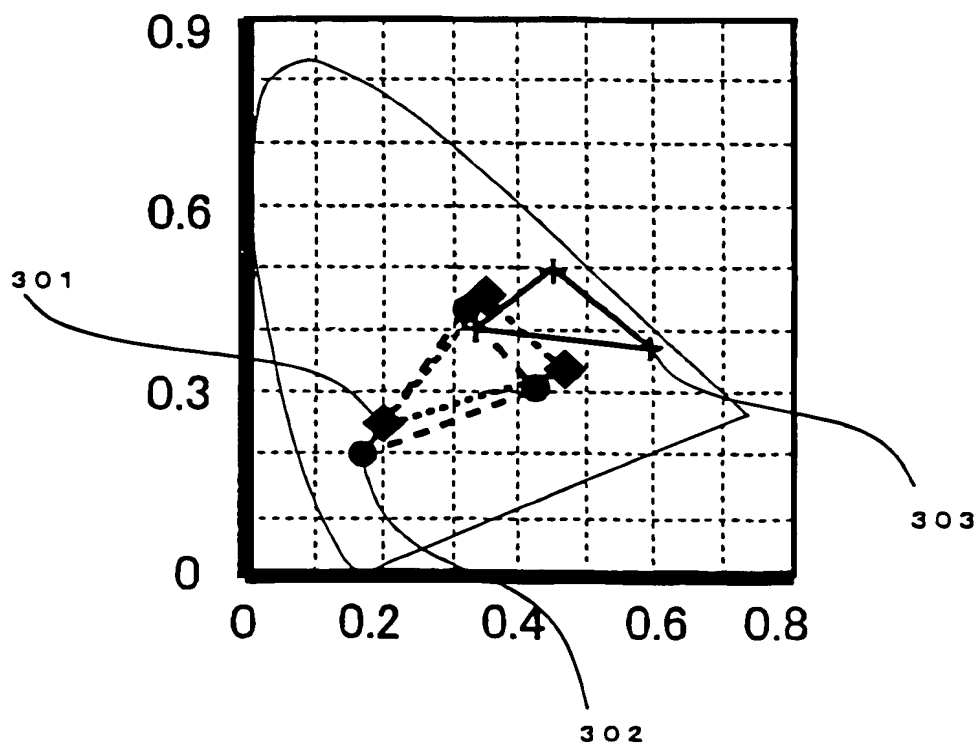


FIG. 4

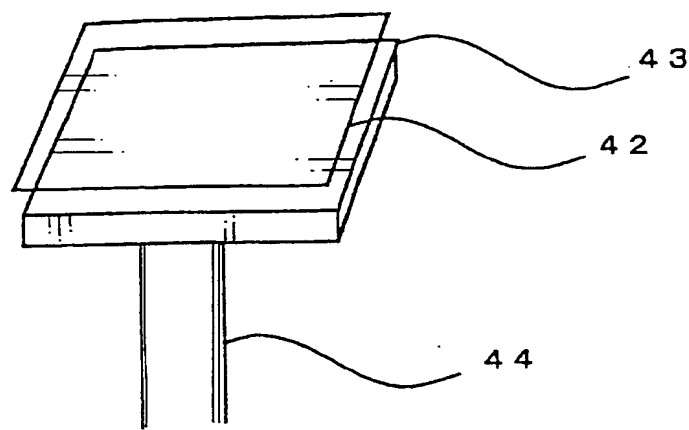


FIG. 5

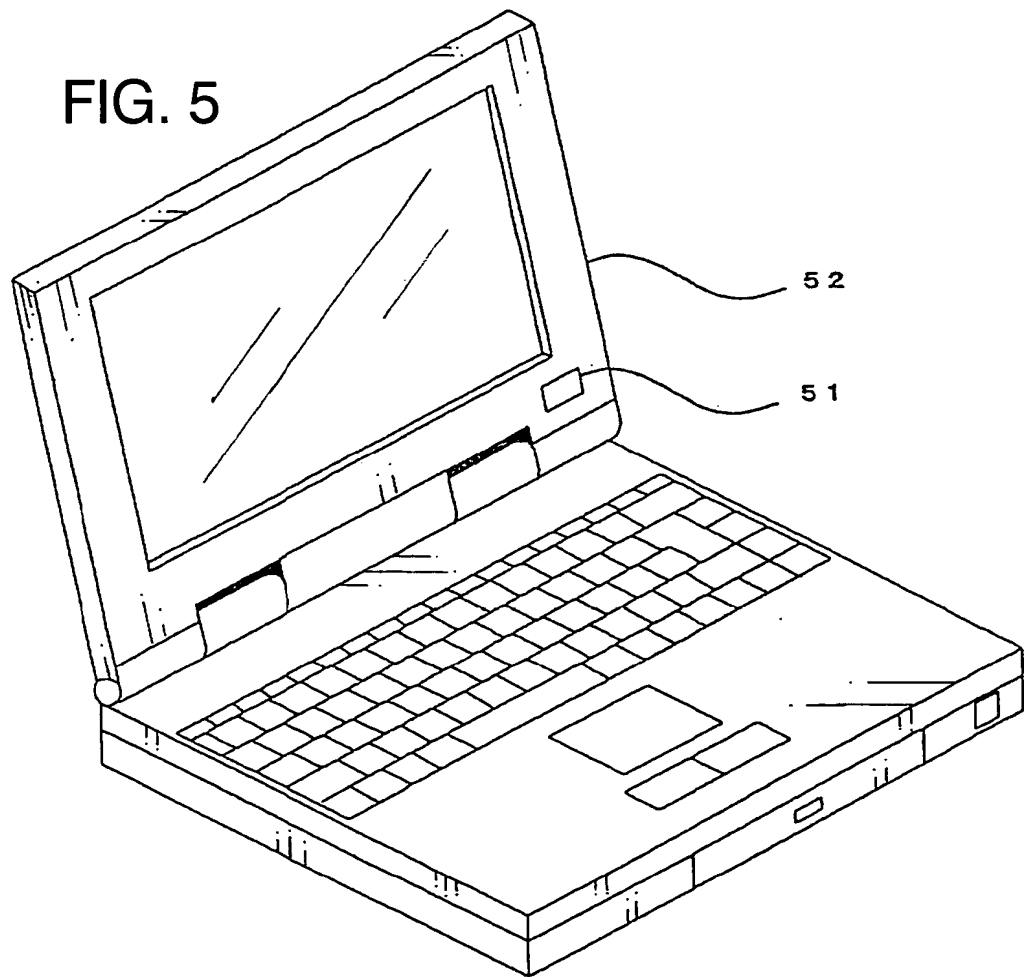


FIG. 6

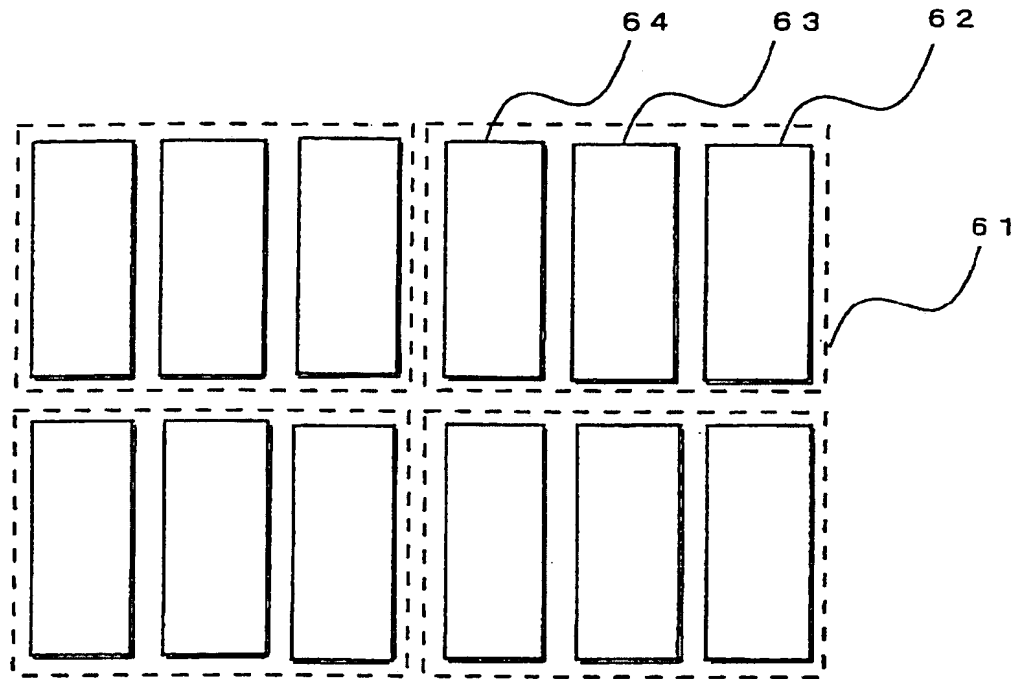


FIG. 7

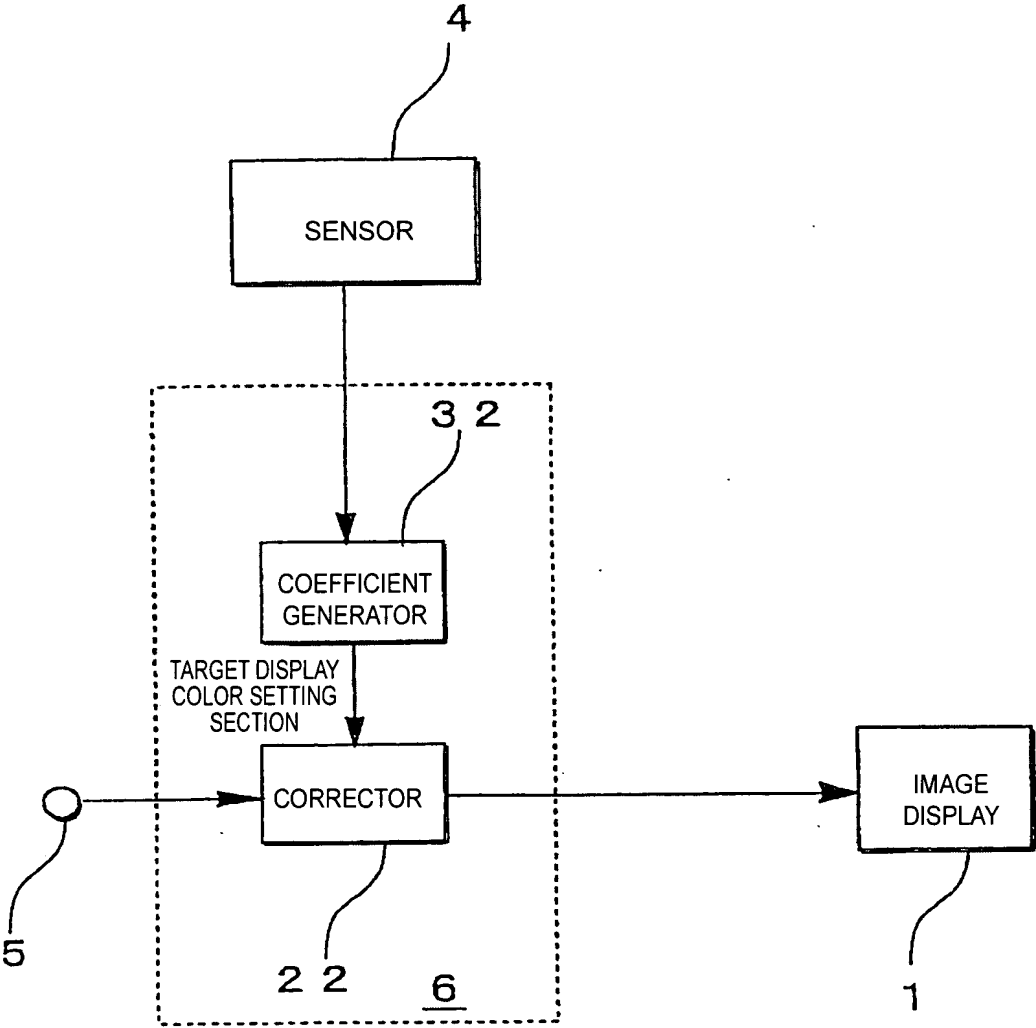


FIG. 8

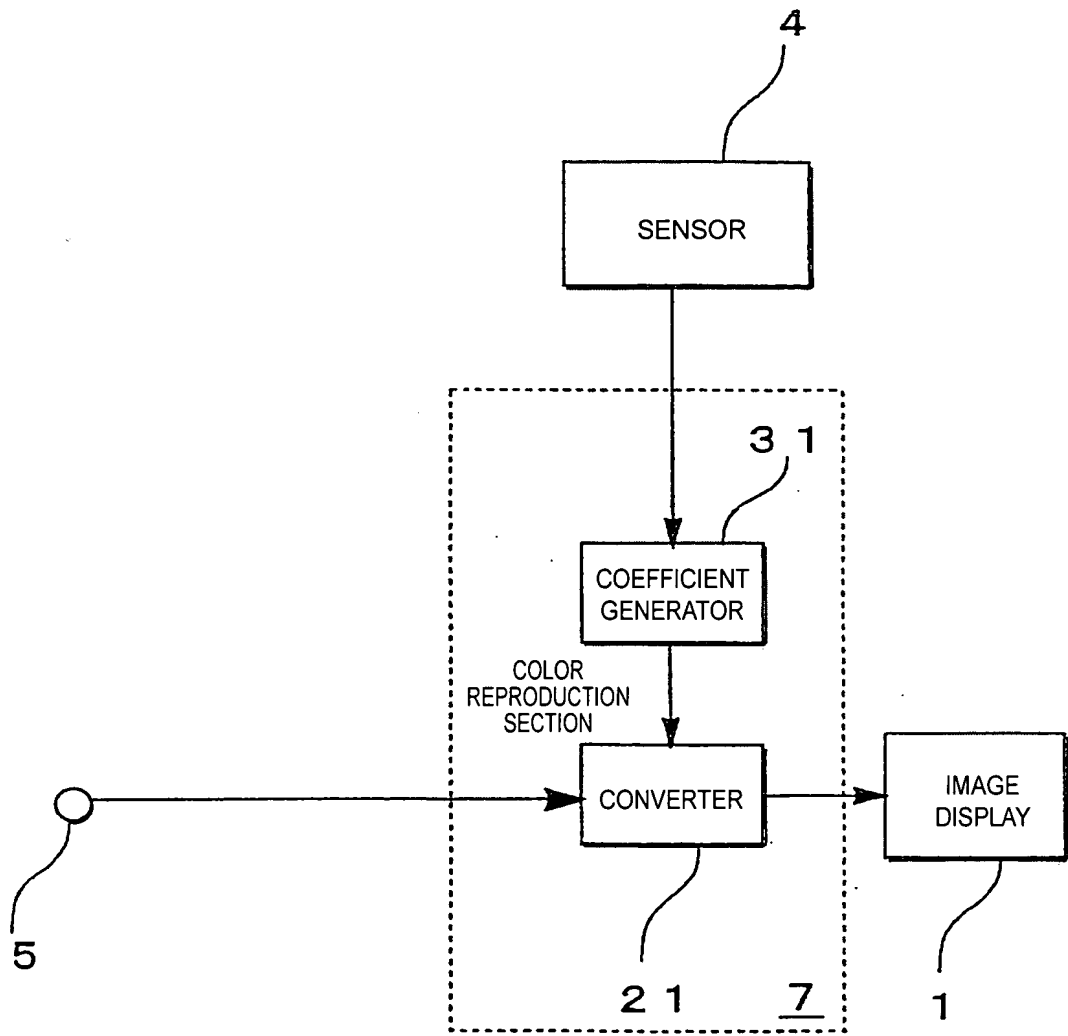


FIG. 9

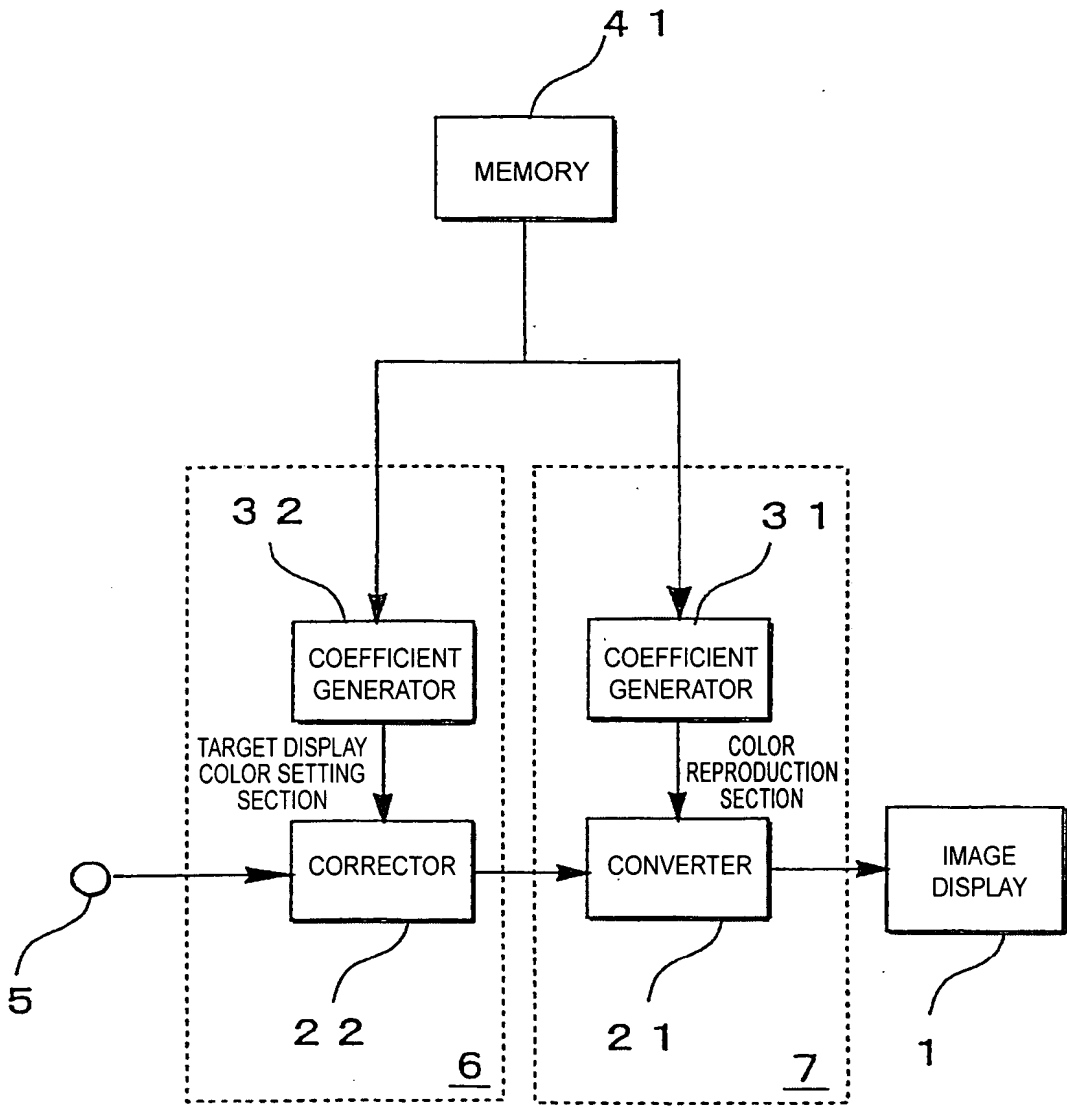


FIG. 10

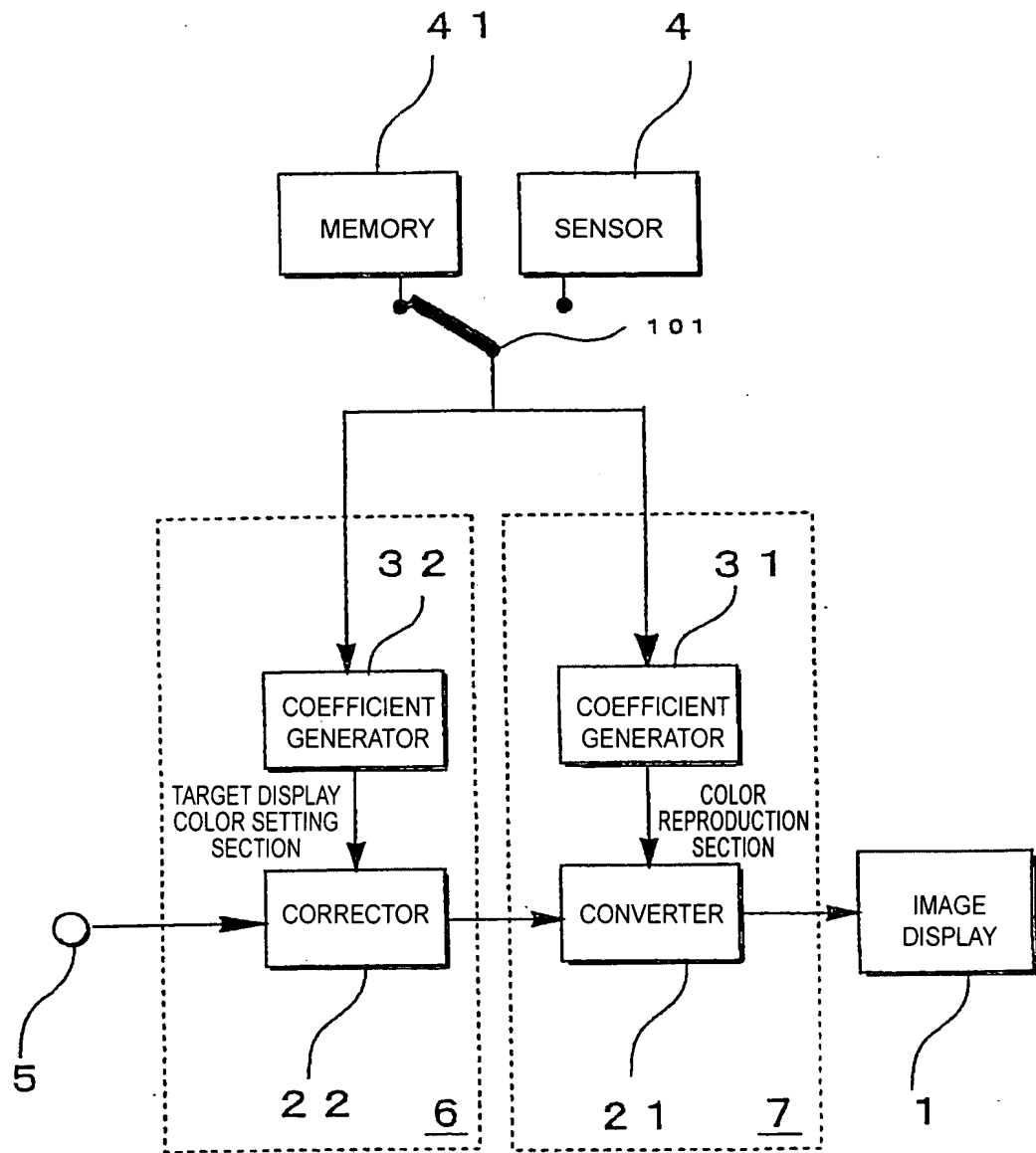


FIG. 11

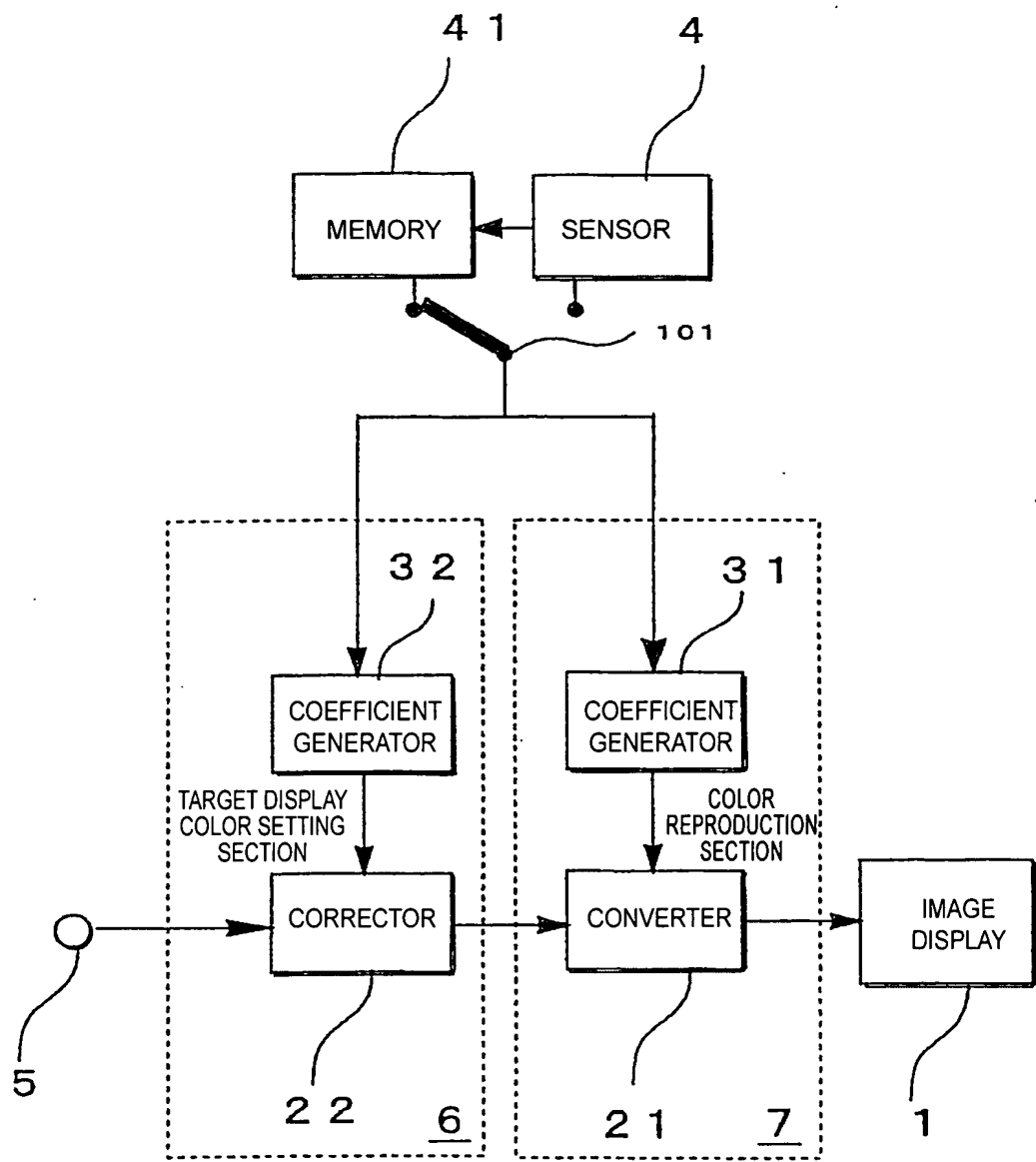


FIG. 12

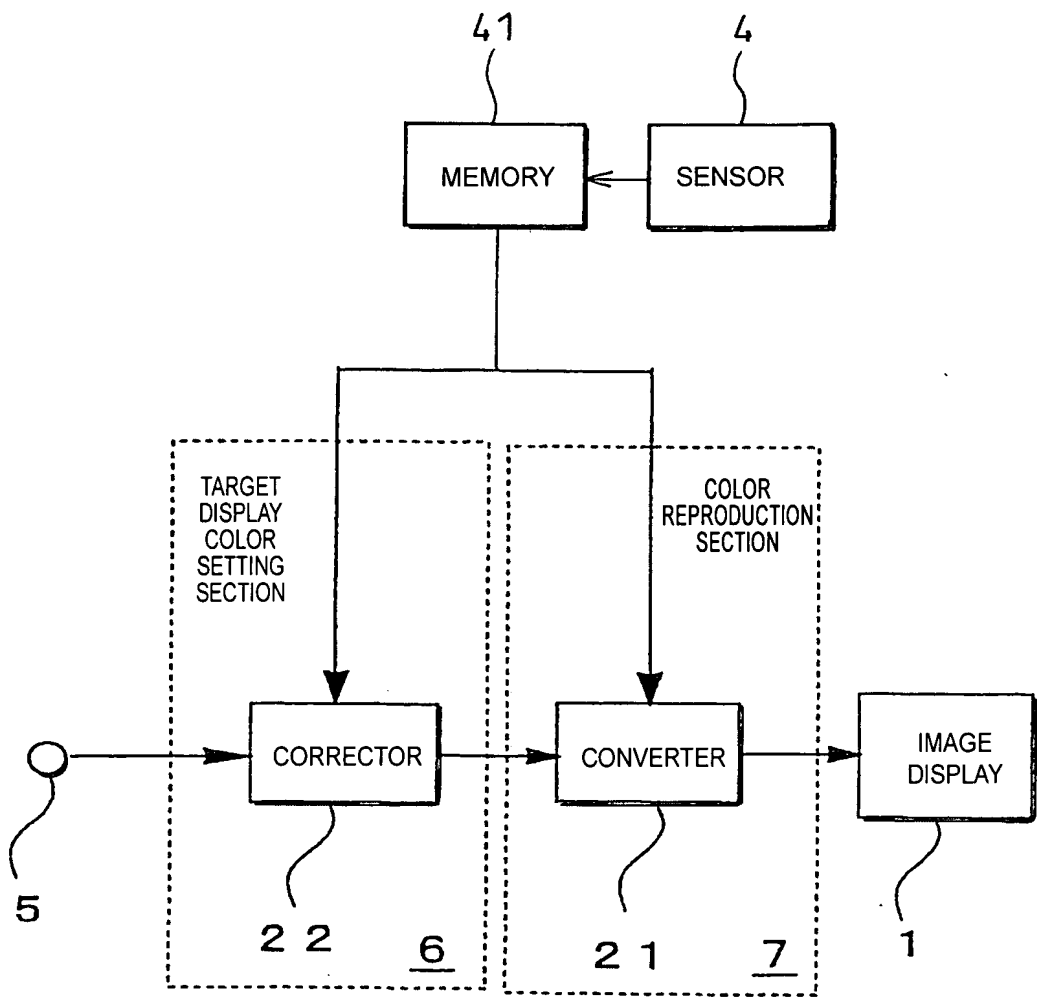


FIG. 13

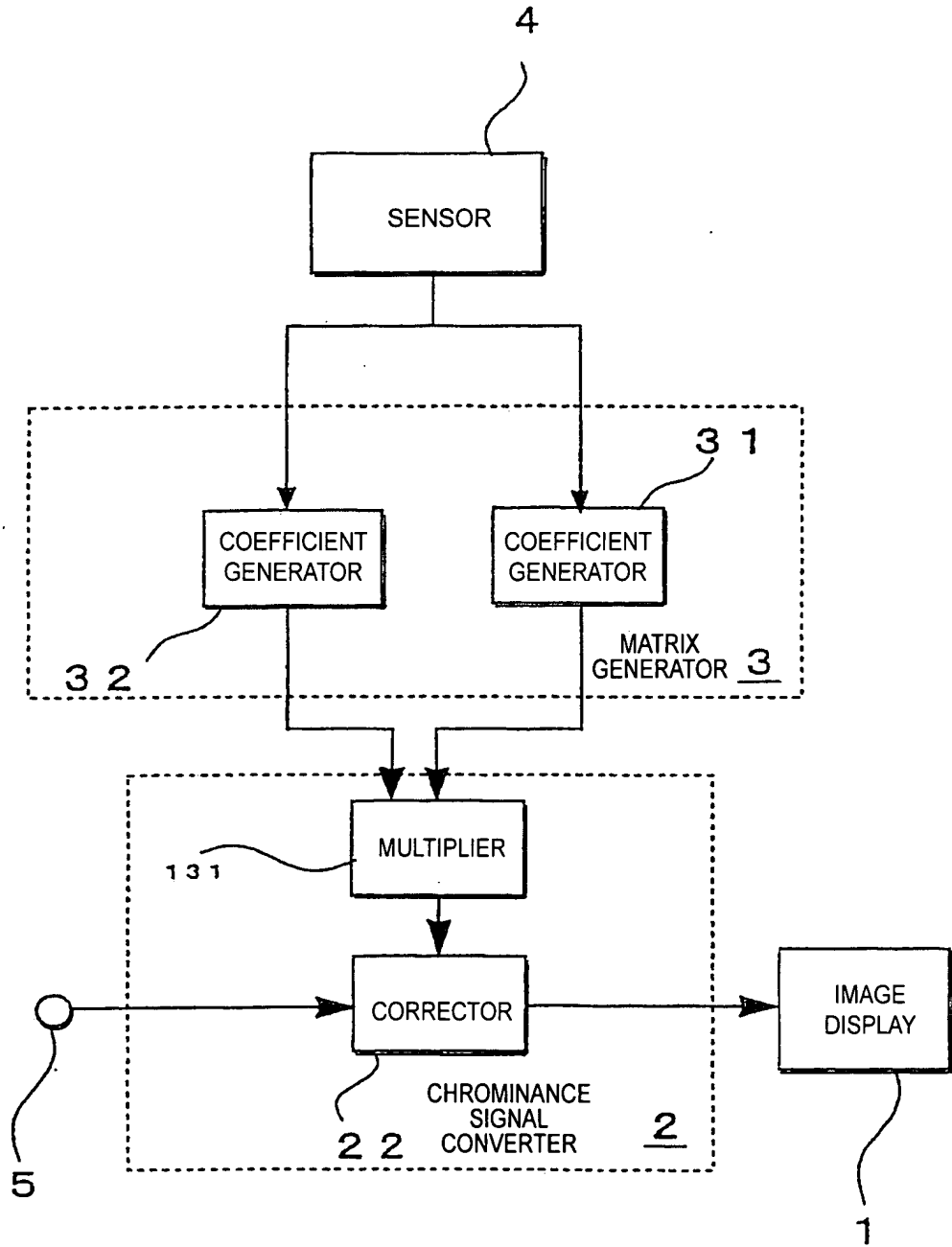


FIG. 14

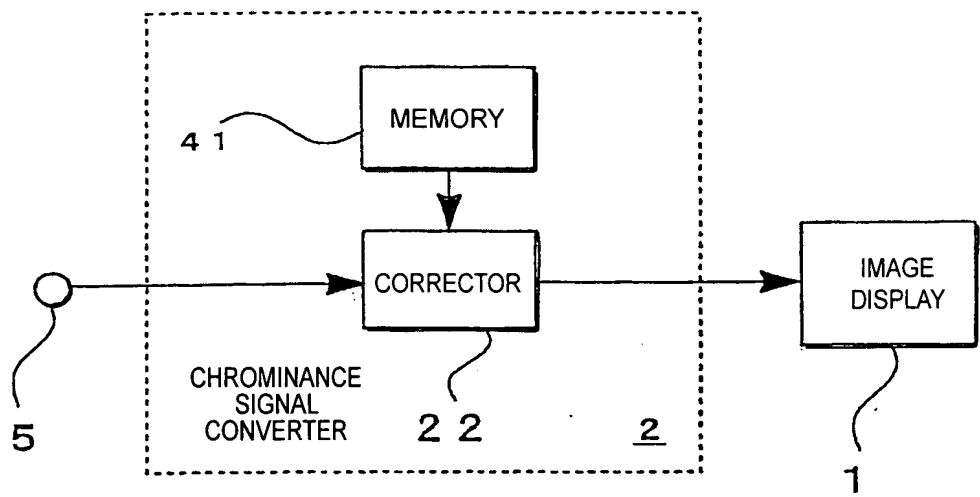


FIG. 15

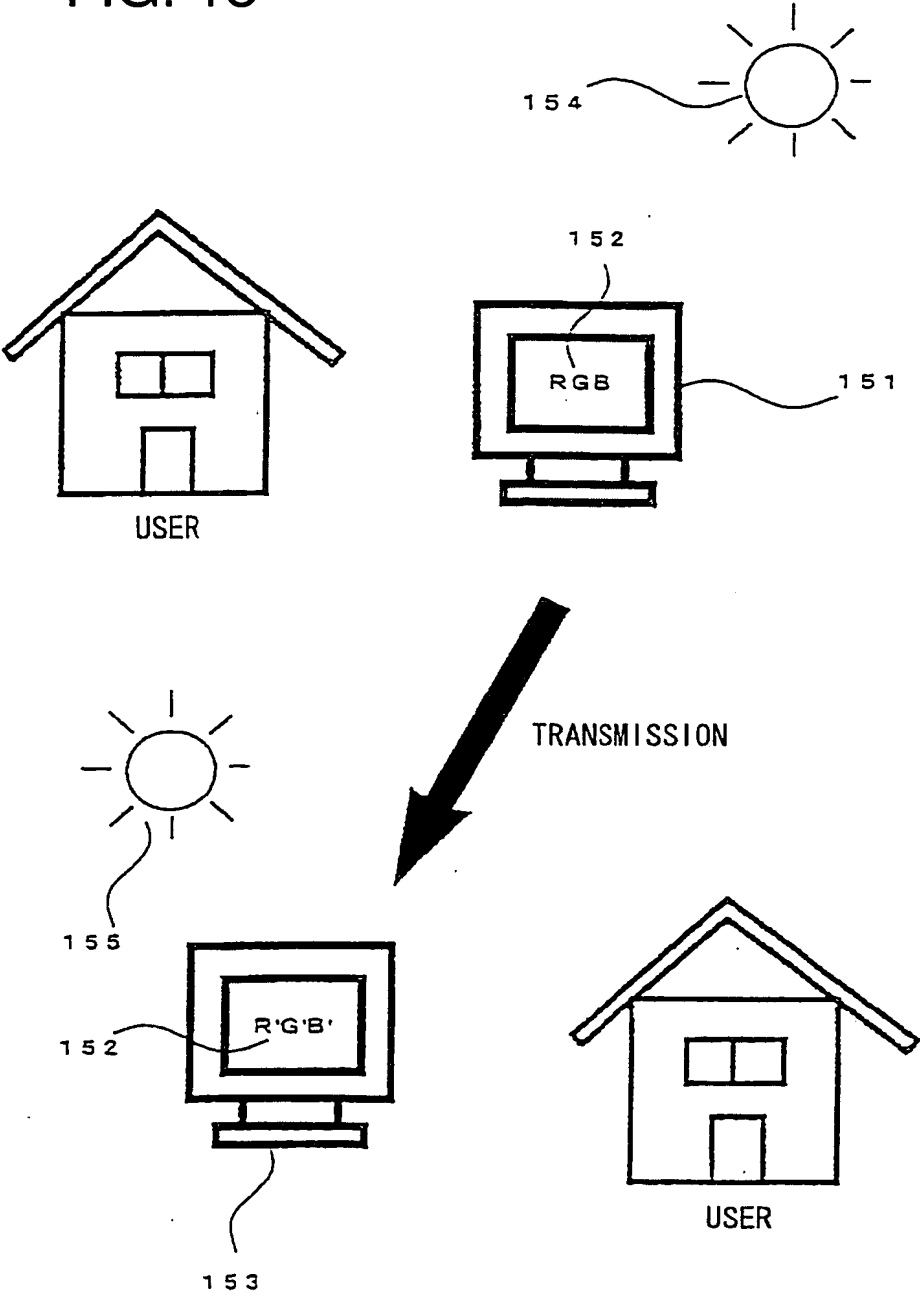


FIG. 16

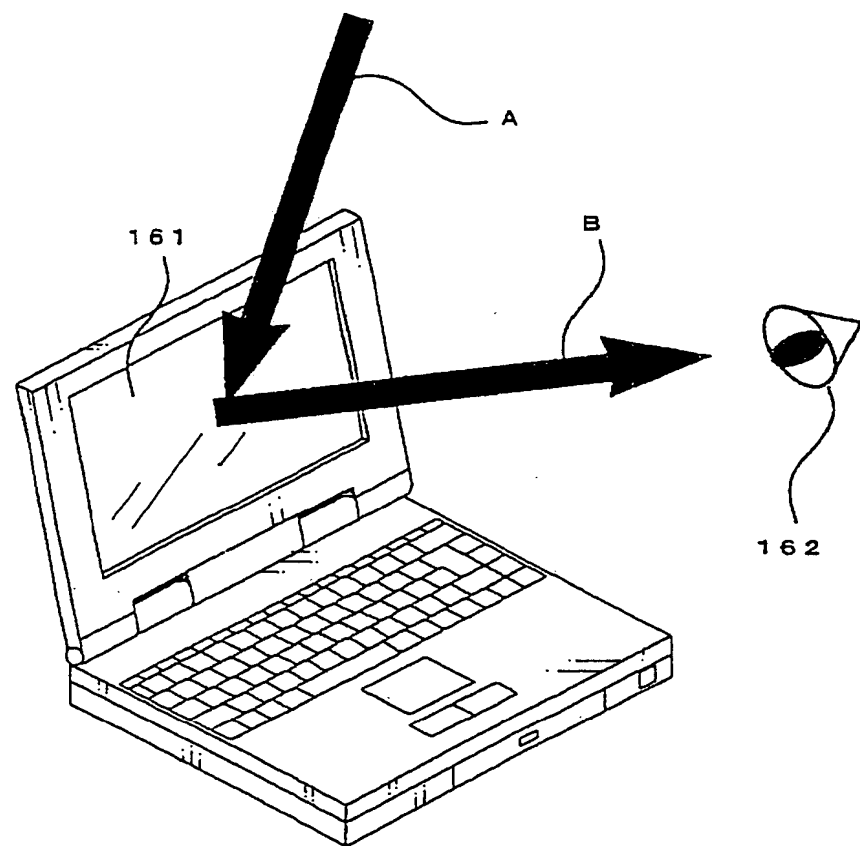


FIG. 17A

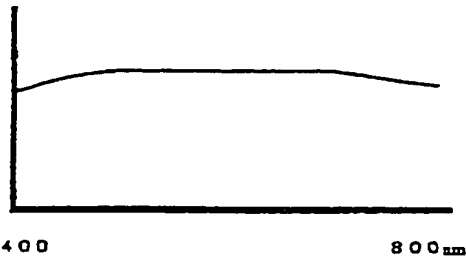


FIG. 17B

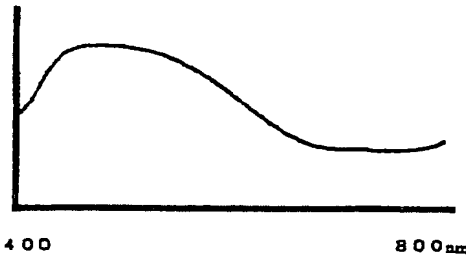


FIG. 17C

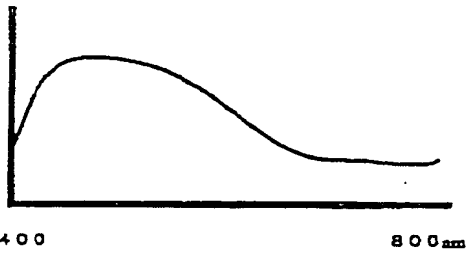


FIG. 17D

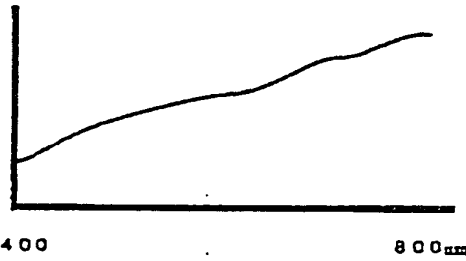


FIG. 17E

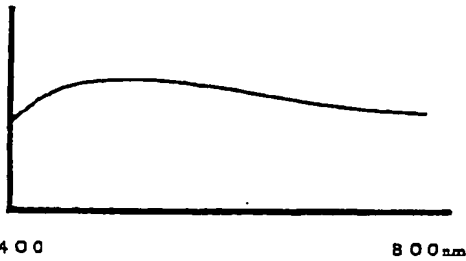


FIG. 18

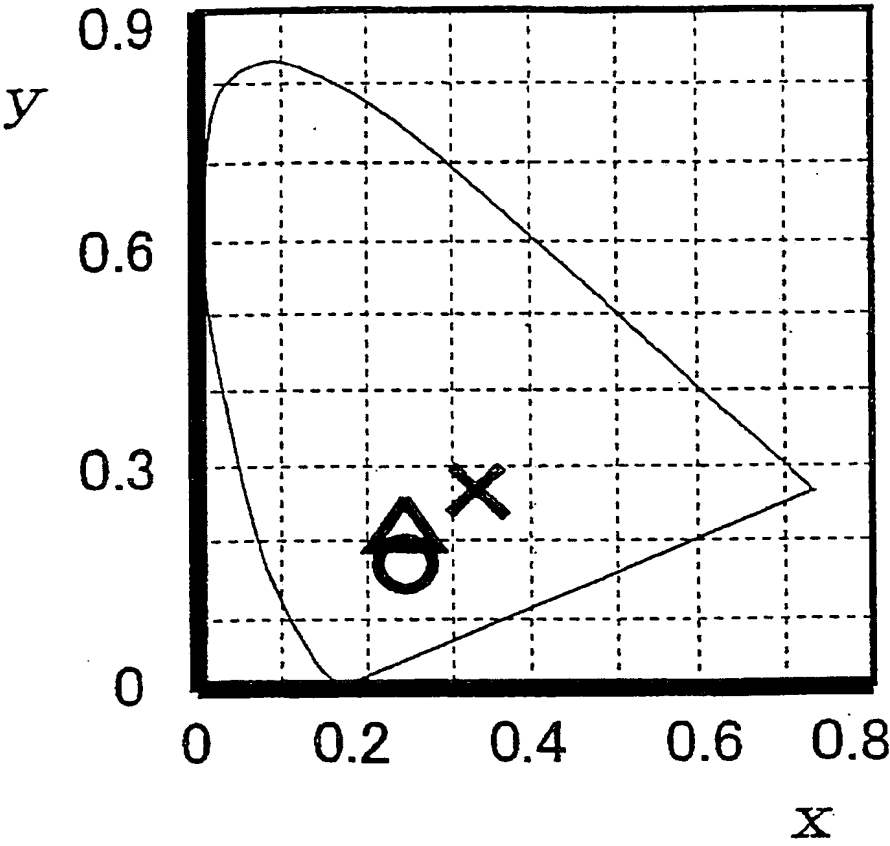


FIG. 19

```

/*****
  transform Program
    for
      colour
      coordinate
*****/
#include <stdio.h>
void
main()
{
    float
d[4][3],a[3][3],b[3],c[3],dd[3],r[3][3],kk[3][3],ss,sss;
    int    i,j,k;

    /* input x & y of RGBW */
    printf("INPUT RGB and White\n");
    printf("Rx Ry Gx Gy Bx By Wx Wy\n");
    scanf("%f %f %f %f %f %f %f %f", &d[0][0],&d[0][1]
                                           , &d[1][0],&d[1][1]
                                           , &d[2][0],&d[2][1]
                                           , &d[3][0],&d[3][1]);

/*
    d[0][0] = 0.67;
    d[0][1] = 0.33;
    d[1][0] = 0.21;
    d[1][1] = 0.71;
    d[2][0] = 0.14;
    d[2][1] = 0.08;
    d[3][0] = 0.31;
    d[3][1] = 0.316;
*/

```

FIG. 20

```

/* calculate z from x & y */
for(i = 0; i < 4; i++){
    if((d[i][0] + d[i][1]) > 1.0){
        d[i][2] = 0.0;
    }
    d[i][2] = 1.0 - d[i][0] - d[i][1];
}

printf("MATRIX\n");
for(i = 0; i < 3; i++){
    printf("%t");
    for( j = 0; j < 3; j++){
        printf("%5.3f%t",d[i][j]);
    }
    printf("%n");
}

```

FIG. 21

```

/* caluculate matrix */
{
    int i1, i2, j1, j2;
    for(i = 0; i < 3; i++){
        i1 = i + 1;
        i2 = i + 2;
        if (i1 > 2) i1 = 0;
        if (i2 > 2) i2 = i2 - 3;
        for(j = 0; j < 3; j++){
            j1 = j + 1;
            j2 = j + 2;
            if (j1 > 2) j1 = 0;
            if (j2 > 2) j2 = j2 - 3;
            a[i][j] = d[i1][j1]*d[i2][j2] - d[i1][j2]*d[i2][j1];
        }
    }
}

/* calculate of BUNBO */
for(i = 0; i < 3; i++){
    b[i] = 0;
    for(j = 0; j < 3; j++){
        b[i] = a[i][j] * d[3][j] + b[i];
    }
}

```

FIG. 22

```

/* MATRIX */
for(i = 0; i < 3; i++){
  for(j = 0; j < 3; j++){
    a[i][j] = a[i][j] / b[i];
    r[i][j] = a[i][j];
    if(i == j){
      kk[i][j] = 1.0;
    } else {
      kk[i][j] = 0.0;
    }
  }
}
/* INVERSE MATRIX */
for(i = 0; i < 3; i++){
  for(j = 0; j < 3; j++){
    dd[j] = a[j][i];
    a[j][i] = 0.0;
  }
  a[i][i] = 1.0;
  for(j = 0; j < 3; j++){
    c[j] = a[i][j] / dd[i];
  }
  for(j = 0; j < 3; j++){
    for(k = 0; k < 3; k++){
      a[j][k] = a[j][k] - c[k]*dd[j];
    }
  }
  for(j = 0; j < 3; j++){
    a[i][j] = c[j];
  }
}

```

FIG. 23

```

/* SEIKIKA */
ss = a[1][0] + a[1][1] + a[1][2];
sss = r[1][0] + r[1][1] + r[1][2];
for(i = 0; i < 3; i++){
  for(j = 0; j < 3; j++){
    a[i][j] = a[i][j] / ss;
    r[i][j] = r[i][j] / sss;
  }
}

```

FIG. 24

```

/* result */
printf("original data\n");
for(i = 0; i < 4; i++){
    printf("%t");
    for( j = 0; j < 3; j++){
        printf("%7.5f  ",d[i][j]);
    }
    printf("\n");
}
printf("MATRIX\n");
for(i = 0; i < 3; i++){
    printf("%t");
    for( j = 0; j < 3; j++){
        printf("%7.5f  ",r[i][j]);
    }
    printf("\n");
}
printf("INVERSE MATRIX\n");
for(i = 0; i < 3; i++){
    printf("%t");
    for( j = 0; j < 3; j++){
        printf("%7.5f  ",a[i][j]);
    }
    printf("\n");
}
for(i = 0; i < 3; i++){
    for(j = 0; j < 3; j++){
        kk[i][j] = a[i][0]*r[0][j] + a[i][1]*r[1][j] + a[i][2]*r[2][j];
    }
}
printf("KAKEZAN\n");
for(i = 0; i < 3; i++){
    printf("%t");
    for( j = 0; j < 3; j++){
        printf("%7.5f  ",kk[i][j]);
    }
    printf("\n");
}
}

```

FIG. 25

